3Ж

УДК: 004.934.1'1

А.В. Жук, С.С. Панфилов
Институт проблем искусственного интеллекта МОН Украины и НАН Украины, г. Донецк
juk@iai.dn.ua, panfilovss@yandex.ru

Алгоритм определения высоты музыкального звука на основе подавления ложных пиков в спектре, построенном для темперированной музыкальной шкалы, и его практическое применение на примере программы музыкального секвенсора

Статья посвящена описанию алгоритма определения частоты основного тона музыкального звука, предполагающего подавление ложных пиков в спектре, построенном для темперированной музыкальной шкалы. Данный алгоритм был реализован в программе музыкального секвенсора с функцией распознавания звукового потока в реальном масштабе времени, описание которого также приводится в данной статье.

Введение

В области цифровой обработки звука задача определения частоты основного тона является довольно важной, а полученные при её решении результаты находят своё применение в задачах идентификации диктора, детектирования и кодирования речи, распознавания музыки. Актуальность данного направления обусловливается спросом на программное обеспечение (ПО) по распознаванию и цифровой обработке звуковых данных. Среди прочих, интерес к такому ПО проявляется и со стороны звукозаписывающих студий, которые занимаются микшированием музыкального материала, восстановлением звуковых фонограмм [1]. Так, в рамках задачи реставрации сведения о частоте основного тона и длительности звукового объекта позволяют:

1) классифицировать компоненты спектра не только как полезные, но и как относящиеся к конкретному звуковому объекту;

2) на основе априорных знаний или эвристических моделей синтезировать утраченные компоненты (обертоны) звуковых объектов и, при необходимости, управлять соотношением параметров (амплитуд, фаз).

Технологии определения частоты основного тона находят свое применение и в «интеллектуальных» караоке-системах, особенностью которых является возможность расчета оценки качества исполнения песни пользователем.

Целью данной статьи является описание алгоритма определения частоты основного тона, основанного на подавлении ложных пиков за счёт использования сведений об особенностях области применения, а также описание архитектуры программного музыкального секвенсора, использующего разработанный алгоритм для заполнения одноголосных партий в реальном масштабе времени на основе записываемого звукового сигнала.

Определение частоты основного тона музыкального звука

В [2] рассматриваются различные подходы к определению частоты основного тона, в том числе и подход, предусматривающий анализ произведений гармоник с кратными частотами в спектре.

Основным источником ошибок при определении частоты основного тона звука по его спектру служат пики, получаемые на частотах, кратных искомой. Зачастую эти «кратные пики» по мощности превышают искомый, что и приводит в результате к ошибке. На рис. 1 изображен спектр музыкального звука с частотой 260 Hz (нота «До» первой октавы). Правильному значению частоты основного тона соответствует первый слева значимый пик, однако при поиске максимального значения будет выбран второй значимый пик, являющийся «кратным».

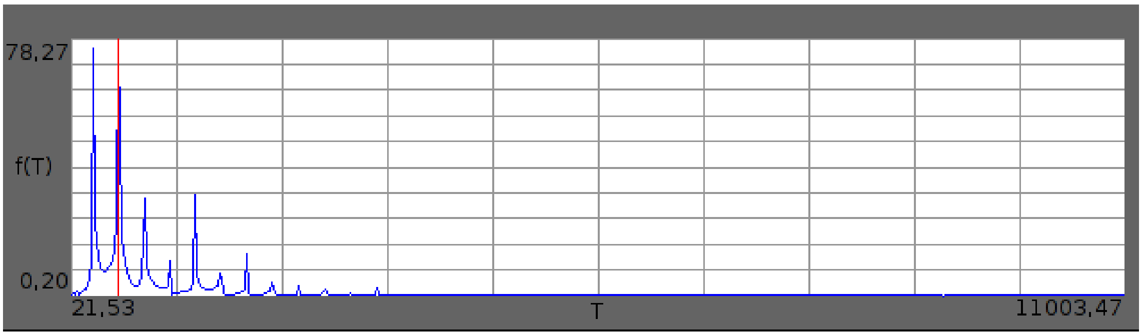


Рисунок 1 – Спектр музыкального звука с частотой 260 Hz

Обычно для получения спектра A используют преобразование Фурье. Следует отметить, что в этом случае полученный набор коэффициентов определяет распределение плотности спектральной энергии по полосам одинаковой ширины Δf_{Φ} :

$$\Delta f_{\Phi} = \frac{f_d}{N},$$

где f_d – частота дискретизации сигнала;

N – размер окна анализа.

В рамках предложенного в [2] подхода предполагается вычисление величин P, представляющих собой логарифмированное произведение мощностей кратных пиков:

$$P[i] = ln \prod_{k=1}^{N/(2i)} A[k \cdot i].$$

Анализируя P вместо A, можно в значительной степени уменьшить влияние кратных пиков на результат. Этот подход послужил основой для алгоритма, описываемого в данной статье.

В современной музыке для описания звукоряда используется темперированная музыкальная шкала, центральную частоту i-го звука которой можно определить по формуле (1) [3]:

$$f_i = f_0 \cdot 2^{k \cdot i/12}, \qquad (1)$$

где f_0 – частота камертона (нота Ля первой октавы – 440Гц);

i – количество полутонов в интервале от искомого звука до f_0 ,

k – коэффициент, зависящий от взаимного расположения звуков:

$$k = \begin{cases} -1, & если \quad f_{звука} < f_0, \\ 1, & иначе. \end{cases}$$

Количество полутонов в интервале, на который отстоит звук с частотой f от ноты «Ля» первой октавы, определяется по формуле (2):

$$i = \log_2\left(\frac{f}{f_0}\right) \cdot \frac{12}{k}. \tag{2}$$

Если исходить из того, что границы между частотными полосами, соответствующими соседним звукам, лежат точно посередине между центральными частотами этих звуков, то ширину $\Delta f_{M\,i}$ полосы частот, отвечающих i-му звуку в рамках темперированной музыкальной шкалы, можно вычислить по формуле (3):

$$\Delta f_{M\,i} = \frac{(f_i + f_{i+1})}{2} - \frac{(f_i + f_{i-1})}{2} = \frac{f_{i+1} - f_{i-1}}{2}. \tag{3}$$

Очевидно, что для разных значений i $\Delta f_{M\,i}$ будет разной. Получается, что хотя анализ спектра Фурье с его полосами одинаковой ширины и может дать достаточно точные результаты для музыкальных звуков, гораздо более предпочтительной является работа со спектром, распределение ширины и центральных частот полос которого соответствует темперированной музыкальной шкале. Значение каждого элемента $Am[i]$ такого спектра можно получить по формуле (4):

$$Am[i] = \sum_{j=\frac{(f_i+f_{i-1})}{2\Delta f_\Phi}}^{\frac{(f_i+f_{i+1})}{2\Delta f_\Phi}} A[j] \tag{4}$$

Спектр *Am*, построенный для ситуации, приведенной на рис. 1, изображен на рис. 2.

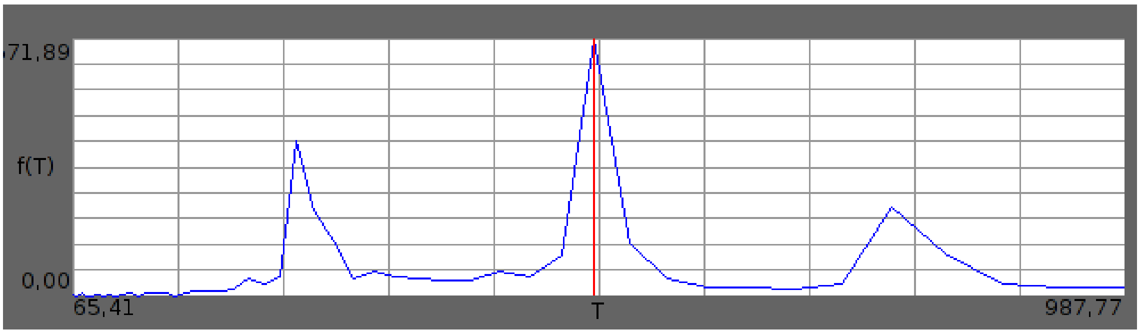


Рисунок 2 – Спектр *Am* музыкального звука с частотой 260 Hz

Между элементами спектра *Am* устанавливаются те же отношения, что и между музыкальными звуками. В темперированной музыкальной шкале частоты звуков, отстоящих на 12 полутонов (одну октаву), отличаются ровно в 2 раза.

Как показывают проведенные исследования, кратные пики могут проявляться на расстоянии до 3-х октав от истинного значения частоты основного тона звука. Использование умножения кратных пиков для усиления искомого значения частоты основного тона несет опасность получения нулевого значения в случае, если один из сомножителей нулевой. Поэтому более перспективным является не усиление мощности искомой

частоты, а гашение мощности кратного пика. Таким образом, в дальнейшем будем рассматривать следующую величину *D* (4) в качестве показателя для определения частоты основного тона музыкального звука:

$$D[i] = Am[i] - \sum_{j=2}^{8} Am[i - 12 \cdot \log_2(j)]. \quad (5)$$

На рис. 3 изображен график показателя *D* для ситуации, соответствующих рис. 1 и 2.

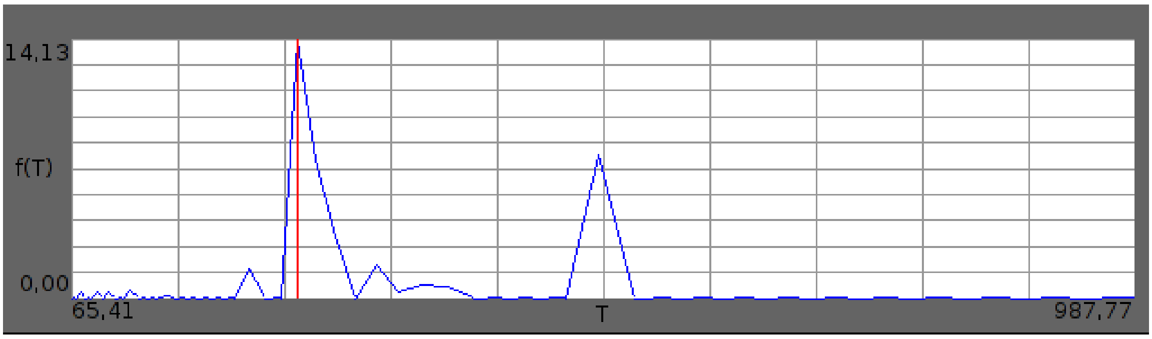


Рисунок 3 – Показатель *D* для музыкального звука с частотой 260 Hz.

Таким образом, процедуру определения значения квазипериода на окне сигнала можно организовать следующим образом:

Шаг 1. Вычисление спектра Фурье *A*.

Шаг 2. Вычисление по (4) спектра *Am*, ширина и центральные частоты полос которого соответствуют темперированной музыкальной шкале.

Шаг 3. Вычисление по (5) показателя *D*.

Шаг 4. Определение индекса *i* максимального значения показателя *D*:

$$i = arg \max_j D[j].$$

Шаг 5. Определение по (1) центральной частоты *i*-й полосы. Данная частота и будет соответствовать частоте основного тона звука на анализируемом окне.

Чтобы построить мелодический контур (определить последовательность квазипериодов) сигнала данный алгоритм можно применять для каждого последовательно выбранного окна. Как показали проведенные исследования, для музыкальных звуков оптимальным является окно анализа размерностью 2048 отсчётов при частоте дискретизации 22050 Hz. Полученный таким образом мелодический контур в задачах обработки записей музыкальных произведений может использоваться в том числе и для определения длительности тех или иных музыкальных звуков, причем в данном случае важным является определение моментов начала и окончания звучания конкретного звука. Поэтому для получения более точных оценок изменения мелодического контура с течением времени необходимо использовать перекрывающиеся окна. В большинстве случаев достаточно половинного перекрытия, однако в зависимости от задачи степень перекрытия окон можно увеличить.

Предложенный алгоритм обладает рядом преимуществ:

1) сравнительно низкая вычислительная сложность, что обеспечивает возможность программной реализации алгоритма для работы в реальном масштабе времени;

2) расчет результатов распознавания производится в единицах стандартизированного музыкального строя, что позволяет эффективно использовать данный алгоритм в программных продуктах, ориентированных на работу с музыкой.

Как показано в [4], точность получаемых результатов определения квазипериода основного тона музыкального звука можно повысить за счёт особой апостериорной обработки мелодического контура, хотя такая обработка и сделает невозможной работу алгоритма в реальном масштабе времени.

Музыкальный секвенсор с возможностью определения частоты основного тона записываемого звука

Пользовательский интерфейс программы

Разработанный алгоритм нашел свое практическое применение при разработке программного музыкального секвенсора.

Основной задачей любого (программного или аппаратного) секвенсора является работа с музыкальными партиями в рамках стандарта MIDI. На рис. 4 представлено главное окно программы с загруженным MIDI-файлом.

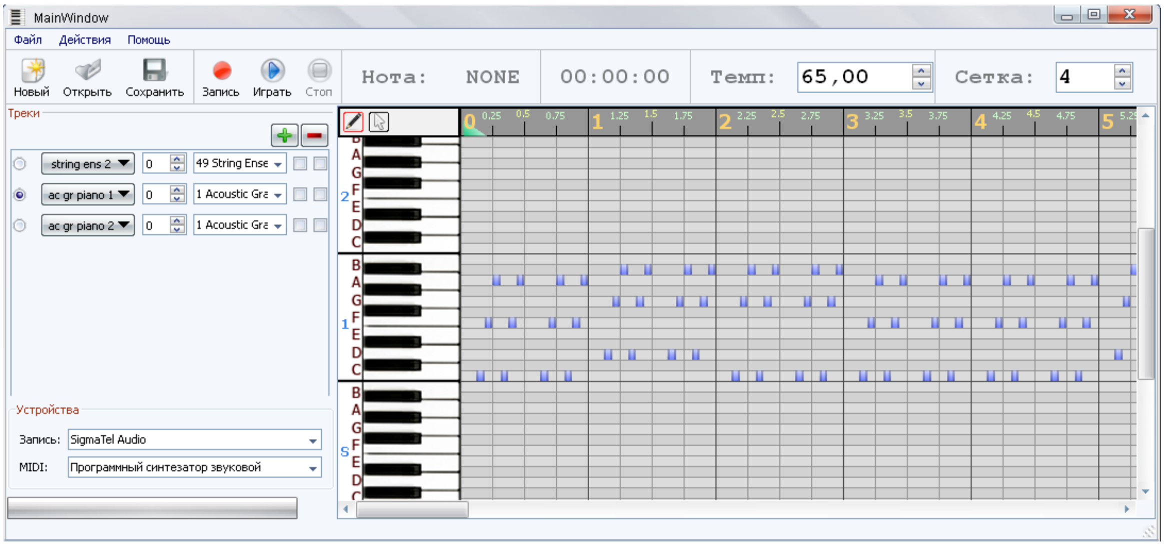


Рисунок 4 – Главное окно музыкального секвенсора.

По объему функциональных возможностей разработанный секвенсор можно отнести к классу средних.

Программа предоставляет следующие возможности:

1) загрузка/сохранение MIDI-файлов с поддержкой всех типов MIDI-сообщений;
2) графическое отображение содержимого MIDI-файла в потрековом режиме;
3) управление треками:
   – добавление/удаление;
   – смена канала;
   – смена инструмента;
   – сольный режим.
4) изменение темпа воспроизведения;
5) воспроизведение партий: общий и сольный режим;
6) редактирование треков;
7) распознавание частоты основного тона, поступающего с устройства захвата;
8) выбор устройств захвата и воспроизведения.

В левой части окна расположен менеджер треков, который дает пользователю доступ к управлению треками. Под менеджером треков находятся списки устройств захвата звука и воспроизведения MIDI-мелодий.

Большую часть окна занимает пианоролл – инструмент для графического отображения и редактирования MIDI-треков. Пианоролл оснащен следующими возможностями:

1) графическое отображение и редактирование MIDI-событий событий начала и окончания звучания ноты в виде прямоугольников на временной оси;
2) масштабирование области вывода;
3) изменение разрядности сетки;
4) выделение элементов;
5) одиночное и групповое перемещение/удаление элементов;
6) изменение длительности нот.

Панель инструментов содержит в себе весь требуемый набор действий: создать новый MIDI-файл, открыть, сохранить, старт захвата данных с микрофона с последующим определением частоты основного тона, остановка захвата/воспроизведения (в зависимости от текущей операции). Панель инструментов также оснащена дополнительными информационными виджетами для отображения распознанной ноты, времени воспроизведения/записи, элементами управления темпом мелодии и размерностью сетки пианоролла.

Объектная модель секвенсора

Секвенсор разработан с использованием кроссплатформенной библиотеки QT 4.x [5], которая распространяется под лицензией LGPL. Базовым языком для QT является C++.

В процессе разработки программы авторы придерживались объектно-ориентированного стиля программирования. На рис. 5 приведена диаграмма основных классов.

Класс MainWindow реализует взаимодействие всех программных модулей между собой: открытие/сохранение, создание новых MIDI-файлов, отображение/редактирование MIDI-сообщений нот, запись, воспроизведение, определение частоты основного тона, управление треками.

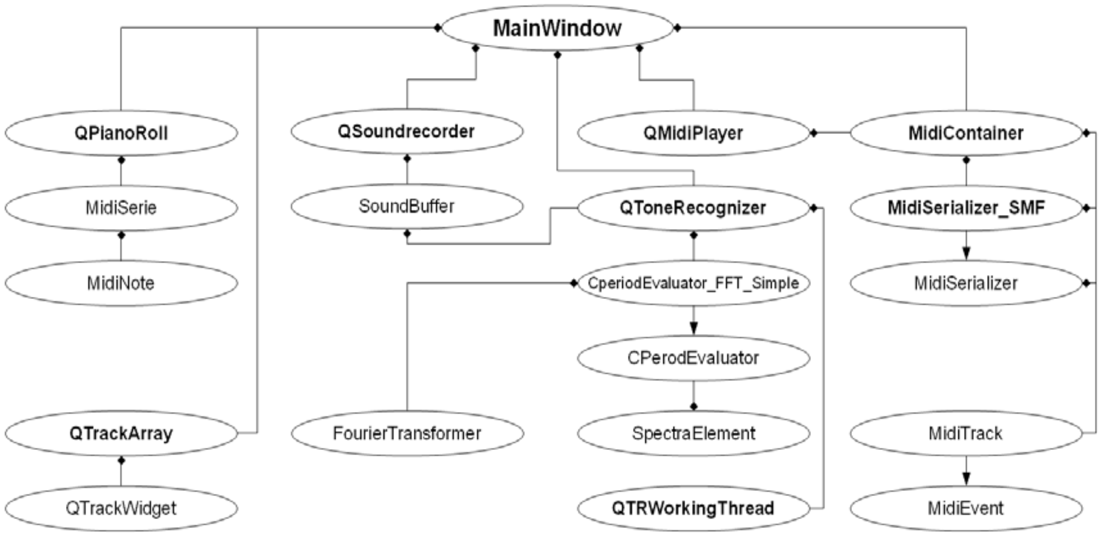


Рисунок 5 – Диаграмма классов музыкального секвенсора

QPianoRoll – класс, реализующий графический редактор партий. Для хранения партий используется вспомогательный класс MidiSerie (массив структур типа MidiNote).

QSoundRecorder реализует процесс захвата данных с аудиоустройства посредством библиотеки BASS [6]. На уровне событий производится отправка звуковых буферов в модуль распознавания.

QToneRecognizer – класс, в котором содержится реализация процесса определения частоты основного тона в отдельном потоке с помощью QTRWorkingThread, который берет на себя роль вычислителя. Фактически разработанный алгоритм определения частоты основного тона реализован в QTRWorkingThread.

Класс MidiContainer предназначен для хранения содержимого загружаемых и сохраняемых MIDI-файлов.

Загрузка и сохранение MIDI-файлов реализована в классе MidiSerializer_SMF.

Классы MidiSerializer_SMF и MidiContainer реализованы в соответствии с ранее разработанной архитектурой компьютерной системы работы со звуком [7].

Класс QMidiPlayer реализует процесс взаимодействия с MIDI-устройствами (выбор устройства, воспроизведение).

Пример работы модуля определения частоты основного тона

Особенностью разработанного секвенсора является возможность определения частоты основного тона. Определение может производиться с любого устройства захвата аудиоданных: микрофон, линейный вход, системный микшер, аналоговый или программный синтезатор и т. п. Возможность выбора устройства записи предоставляется интерфейсом программы. По нажатию кнопки «Запись» будет начат захват аудиоданных, параллельно с которым будет выполняться определение частоты основного тона.

На рисунке 6 приведен фрагмент аудиосигнала, который будет воспроизведен при помощи MIDI-синтезатора.

На рис. 7 приведен результат распознавания.

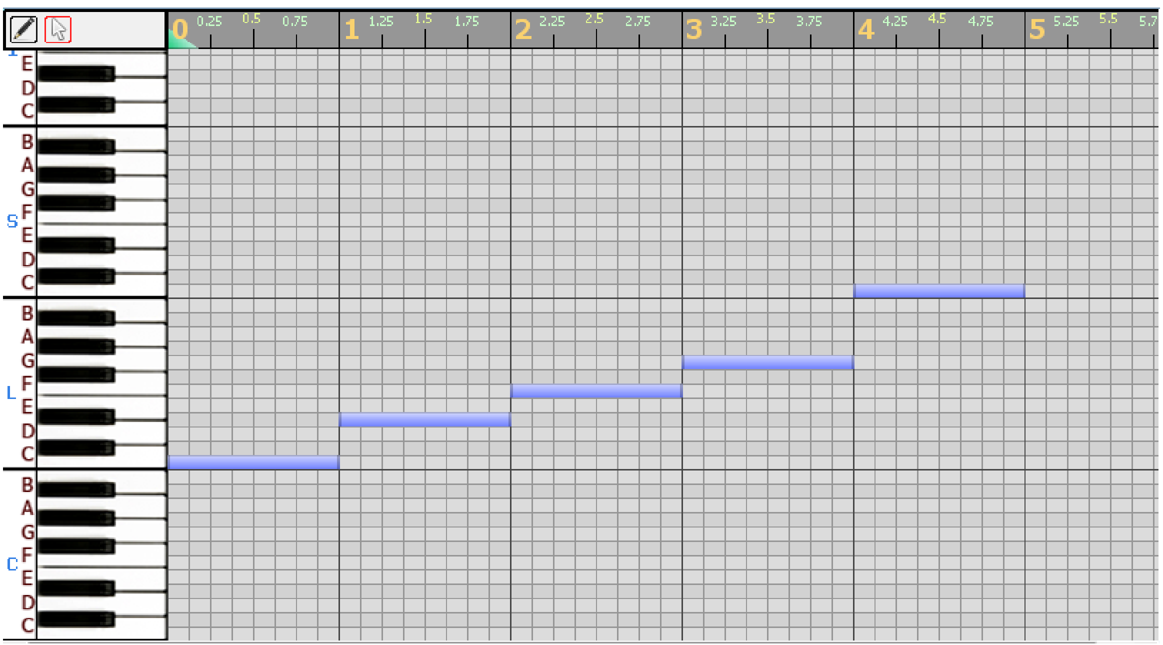


Рисунок 6 – Исходная последовательность звуков

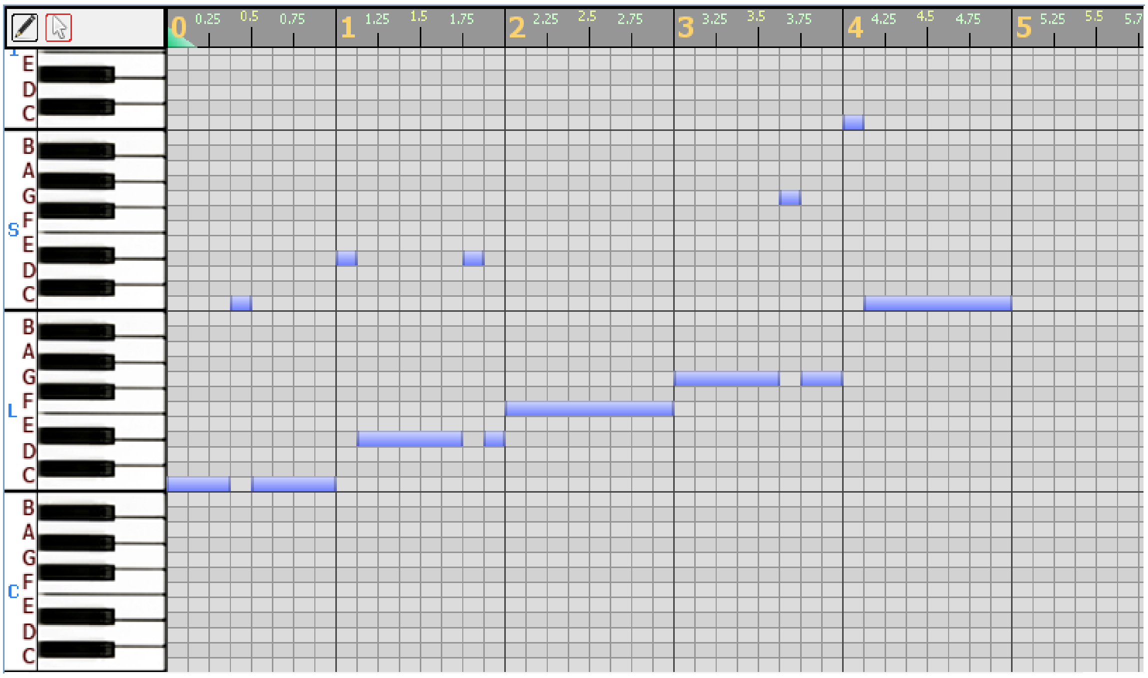


Рисунок 7 – Распознанная последовательность звуков

Несмотря на наличие ошибок распознавания, результат является достаточно правдоподобным. Продолжительное тестирование распознавателя показало, что в основном, ошибки заключаются в сдвиге правильного результата на одну или более октав вверх.

Выводы

В данной работе был изложен алгоритм определения частоты основного тона с использованием музыкальной шкалы на основе метода подавления ложных пиков в спектре входного сигнала. Данный алгоритм применен в разработанном программном музыкальном секвенсоре. Небольшая вычислительная сложность позволила реализовать работу алгоритма в режиме реального времени. Отмечен недостаток: наличие мелких вкраплений (отклонений на одну или несколько октав). Существуют методы для решения данной проблемы, но, к сожалению, они являются непригодными для работы в реальном времени.

В работе также изложена архитектура разработанного секвенсора. В ней нашли продолжение элементы предложенной ранее распределенной архитектуры компьютерной системы для работы со звуком. Это позволит без затруднений использовать существующие результаты в дальнейших работах.

Литература

1. Алдошина И Компьютерное распознавание высоты и тембра звука. Электронный журнал «Звукорежиссер», 2005. – Режим доступа: URL: http://rus.625-net.ru/audioproducer/2005/01/aldo.htm
2. Рабинер Л.Р., Шафер Р.В. Цифровая обработка речевых сигналов / Л.Р. Рабинер, Р.В. Шафер ; пер. с англ. ; под ред. М.В. Назарова и Ю.Н. Прохорова. – М. : Радио и связь, 1981. – 496 с., ил.
3. Equal temperament. Режим доступа : URL: http://en.wikipedia.org/wiki/Equal_temperament
4. Жук А.В. Распознавание последовательности звуков одноголосной мелодии по высоте / А.В. Жук // Искусственный интеллект. – 2006. – № 2.
5. QT 4.x.Режим доступа : URL: http://qt.nokia.com/products

6. BASS Режим доступа : URL: – http://www.un4seen.com/
7. Жук А.В. Разработка элементов расширенной архитектуры компьютерной системы работы со звуком / А.В. Жук, С.С. Панфилов // Искусственный интеллект. – 2010. – № 2. – С. 305-312.

Literatura

1. Aldoshina I, Komp'yuternoe raspoznavanie vysoty i tembra zvuka. Elektronnyy zhurnal «Zvukorezhisser», 2005. Sposob dostupa: URL: http://rus.625-net.ru/audioproducer/2005/01/aldo.htm
2. Rabiner L.R., Shafer R.V. Tsifrovaya obrabotka rechevykh signalov: Per. s angl./Pod red. M.V. Nazarova i Yu.N. Prokhorova- M.: Radio i svyaz', 1981.- 496 s., il.
3. Equal temperament. Sposob dostupa: URL: http://en.wikipedia.org/wiki/Equal_temperament
4. Zhuk A.V., Raspoznavanie posledovatel'nosti zvukov odnogolosnoy melodii po vysote // Iskusstvennyy intellekt. – 2006. – № 2.
5. QT 4.x. Sposob dostupa: URL: http://qt.nokia.com/products
6. BASS. Sposob dostupa: URL: – http://www.un4seen.com/
7. Zhuk A.V., Panfilov S.S. Razrabotka elementov rasshirennoy arkhitektury komp'yuternoy sistemy raboty so zvukom // Iskusstvennyy intellekt. – 2010, № 2, S. 305-312.

Жук О.В., Панфілов С.С.

Алгоритм визначення висоти музичного звуку на основі зниження рівня хибних піків в спектрі, побудованому для темперованої музичної шкали, та його практичне застосування на прикладі програми музичного секвенсора

Статтю присвячено опису алгоритму визначення основного тону музичного звуку, що передбачає зниження рівня хибних піків в спектрі, побудованому для темперованої музичної шкали. Даний алгоритм був реализований в програмі музичного секвенсора з функцією розпізнавання звукового потоку в реальному масштабі часу, опис якого також наведено в даній статті.

Zhuk A., Panfilov S.

An Algorithm for Pitch Extraction Using Dumping of False Peaks in Equal-Temperament-Based Spectra and it's Employment in the Musical Sequencer Application

The article is devoted to description of an algorithm for pitch extraction, which implies dumping of false peaks in equal-temperament-based spectra. The algorithm was employed within musical sequencer application with function for real time recognition of sound stream. The application structure is also described in this article

Статья поступила в редакцию 15.06.2011.